In the Claims:

1. (Currently Amended) A clock ratio data synchronizer, the synchronizer comprising a plurality of flip flops, each flip flop capable of sampling data and outputting data only on an edge of a clock, wherein the synchronizer synchronizes data received by the synchronizer from first clock domain logic at a first clock frequency to a second clock domain logic at a second clock frequency, the first clock domain logic being controlled by a first clock that generates the first clock frequency, the second clock domain logic being controlled by a second clock that generates the second clock frequency, the first and second clocks being synchronized to each other at regular intervals, the synchronizer having a Data In input and a Data Out output, and wherein logic gates of the first clock domain logic that are in a signal path that is coupled to the Data In input are provided by the synchronizer with a first clock cycle as a setup time margin.

2. (Previously Presented) The synchronizer of claim 1, wherein the synchronizer further comprises:

Data In input logic that receives data from the first clock domain logic at an input of a first one of said flip flops, said first one of said flip flops sampling data received at an input thereof when an edge of said first clock is received by said first one of said flip flops, said first one of said flip flops outputting the data sampled thereby from an output thereof when an edge of said first clock is received by said first one of said flip flops;

synchronization logic that receives the data output from said first one of said flip flops, said synchronization logic including a first data path and a second data path, the data output from said first one of said flip flips being gated along the first and second data paths in accordance with one or more timing signals and edges of said first clock such that data from the first and second data paths is output from the synchronization logic in a predetermined order that resembles an order of the data received at the Data In input logic;

Data Out output logic that receives the ordered data from the synchronization logic in said predetermined order and outputs the ordered data from the synchronizer to the second clock domain logic at the second clock frequency, wherein the Data Out output logic includes an output flip flop that receives and samples the ordered data and that outputs the data sampled thereby at said second clock frequency when an edge of said second clock is received by said output flip flop.

3. (Cancelled)

4. (Previously Presented) The synchronizer of claim 1, wherein the synchronizer provides logic gates of the second clock domain logic that are in a signal path that is coupled to the Data Out output of the synchronizer with a second clock cycle as a setup time margin.

5. (Original) The synchronizer of claim 2, wherein the first and second data paths of the synchronization logic each comprise at least one synchronization flip flop that samples the data output from said first one of said first flip flips, each synchronization flip flop sampling the data output from said first one of said first flip flops when an edge of the first clock occurs and outputting the data sampled by the respective synchronization flip flop when an edge of the first clock occurs.

6. (Original) The synchronizer of claim 1, wherein the synchronizer provides a clock skew tolerance margin with respect to the first and second clocks that is equal to at least half of whichever of the first and second clock cycles is shortest in time duration.

7. (Original) The synchronizer of claim 1, wherein the clock frequency of the first clock domain logic is higher than the clock frequency of the second clock domain logic.

8. (Original) The synchronizer of claim 1, wherein the clock frequency of the second clock domain logic is higher than the clock frequency of the first clock domain logic.

9. (Original) The synchronizer of claim 1, wherein a ratio of the first clock frequency to the second clock frequency is N:N-1, where N is an integer greater than or equal to 2.

10. (Original) The synchronizer of claim 1, wherein a ratio of the first clock frequency to the second clock frequency is N-1:N, where N is an integer greater than or equal to 2.

11. (Previously Presented) A clock ratio synchronizer for synchronizing data received by the synchronizer from a first clock domain operating at a first clock frequency to a second clock domain operating at a second clock frequency, the first clock frequency being generated by a first clock, the second clock frequency being generated by a second clock, the synchronizer comprising:

input logic that receives data from the first clock domain, the input logic including an input flip flop that samples data when an edge of said first clock is received by said input flip flop, wherein when an edge of said first clock is received by said input flip flop, said input flip flop outputs a data value corresponding to the data sampled by the input flip flop;

synchronization logic that synchronizes the data received at the input flip flop to the second clock frequency, the synchronization logic comprising one or more synchronization flip flops and one or more multiplexers, at least one of the synchronization flip flops sampling the data value output by said input flip flop when an edge of the second clock occurs, and wherein the data sampled by said at least one of the synchronization flip flops is output therefrom when an edge of the second clock occurs, and wherein each of the multiplexers receives a respective control signal that causes the respective multiplexer to output one of two data values received thereby when the respective control signal has a certain value; and

output logic that receives data values output from said synchronization logic and outputs data therefrom at the second clock frequency into the second clock domain, the output logic comprising at least one output flip flop that samples data on an edge of said second clock and that outputs the data sampled by the second flip flop on an edge of said second clock.

12. (Original) The synchronizer of claim 11, wherein the first clock frequency is greater than the second clock frequency.

13. (Original) The synchronizer of claim 11, wherein the second clock frequency is greater than the first clock frequency.

14. (Previously Presented) The synchronizer of claim 11, wherein the first clock domain comprises first clock domain logic and wherein logic gates of the first clock domain logic that are in a signal path of the input logic of the synchronizer have a first clock cycle as a setup time margin.

15. (Previously Presented) The synchronizer of claim 14, wherein the second clock domain comprises second clock domain logic and wherein logic gates of the second clock domain logic that are in a signal path of the output logic of the synchronizer have a second clock cycle as a setup time margin.

16. (Original) The synchronizer of claim 11, wherein the synchronizer provides a clock skew margin tolerance with respect to the first and second clocks that is at least half of whichever of the first and second clock cycles is shortest in time duration.

17. (Original) The synchronizer of claim 11, wherein a ratio of the first clock frequency to the second clock frequency is N:N-1, where N is an integer greater than or equal to 2.

18. (Original) The synchronizer of claim 11, wherein a ratio of the first clock frequency to the second clock frequency is N-1:N, where N is an integer greater than or equal to 2.

19. (Currently Amended) A clock ratio data synchronizer for synchronizing data received by the synchronizer from a first clock domain operating at a first clock frequency to a second clock domain operating at a second clock frequency, the first clock frequency being generated by a first clock, the second clock frequency being generated by a second clock, the synchronizer comprising:

a first input flip flop, the first input flip flop receiving data from the first clock domain at an input of the first input flip flop and sampling the received data only on an edge of said first clock, the first flip flop outputting the sampled data therefrom only on an edge of said first clock;

an output flip flop, the output flip flop receiving the data output from the input flip flop at an input of the first output flip flop and sampling the data received by the output flip flop only on an edge of said second clock, the output flip flop outputting the data sampled by the output flip flop only on an edge of said second clock, and wherein logic gates of the first clock domain logic that are in a signal path that is coupled to the input of the first input flip flop are provided by the synchronizer with a first clock cycle as a setup time margin, and wherein the first and second clocks are synchronized to each other at regular intervals.

20. (Currently Amended) A method for synchronizing data received by a synchronizer from a first clock domain operating at a first clock frequency to a second clock domain operating at a second clock frequency, the first clock frequency being generated by a first clock, the second clock frequency being generated by a second clock, the method comprising the steps of:

receiving data from the first clock domain at an input of an input flip flop;

sampling the received data only on an edge of said first clock;

outputting data from an output of said input flip flop only on an edge of said first clock;

receiving the data output from the input flip flop at an input of an output flip flop;

sampling the data received by the output flip flop only on an edge of said second clock; and

outputting data from an output of said output flip flop only on an edge of said second clock, , wherein the first and second clocks are synchronized to each other at regular intervals, and wherein logic gates of the first clock domain logic that are in a signal path that is coupled to the input of the first input flip flop are provided by the synchronizer with at least a first clock cycle as a setup time margin.

21. (Currently Amended) A clock ratio data synchronizer, the synchronizer comprising a plurality of flip flops, each flip flop capable of sampling data and outputting data only on an edge of a clock, wherein the synchronizer synchronizes data received by the synchronizer from first clock domain logic at a first clock frequency to a second clock domain logic at a second clock frequency, the first clock domain logic being controlled by a first clock that generates the first clock frequency, the second clock domain logic being controlled by a second clock that generates the second clock frequency, and wherein a ratio of the first clock frequency to the second clock frequency is a non-integer, and wherein the first and second clocks are synchronized to each other at regular intervals.

22. (Previously Presented) The clock ration synchronizer of claim 21, wherein a ratio of the second clock frequency to the first clock frequency is a non-integer.

23. (Currently Amended) A method for synchronizing data received by a synchronizer from a first clock domain operating at a first clock frequency to a second clock domain operating at a second clock frequency, the first clock frequency being generated by a first clock, the second clock frequency being generated by a second clock, the method comprising the steps of:

receiving data from the first clock domain at an input of an input flip flop;

sampling the received data only on an edge of said first clock;

outputting data from an output of said input flip flop only on an edge of said first clock;

receiving the data output from the input flip flop at an input of an output flip flop;

sampling the data received by the output flip flop only on an edge of said second clock; and

outputting data from an output of said output flip flop only on an edge of said second clock, and wherein the first and second clocks are synchronized to each other at regular intervals, wherein a ratio of the first clock frequency to the second clock frequency is a non-integer.

24. (Previously Presented) The method of claim 23, wherein a ratio of the second clock frequency to the first clock frequency is a non-integer.